Comparison of solvers performance when solving the 3D Helmholtz elastic wave equations over the Hybridizable Discontinuous Galerkin method

Marie Bonnasse-Gahot, Henri Calandra, Julien Diaz, Stephane Lanteri

Comparison of solvers performance when solving the 3D Helmholtz elastic wave equations over the Hybridizable Discontinuous Galerkin method

M. Bonnasse-Gahot[1,2], H. Calandra[3], J. Diaz[1] and S. Lanteri[2]

[1] *INRIA Bordeaux-Sud-Ouest, team-project Magique 3D*
[2] *INRIA Sophia-Antipolis-Méditerranée, team-project Nachos*

[3] *TOTAL Exploration-Production*

October 27, 2016

Motivations

Imaging methods

- Full Wave Inversion (FWI) : **inversion process** requiring to solve many forward problems

Motivations

Imaging methods

- Full Wave Inversion (FWI) : **inversion process** requiring to solve many forward problems

Seismic imaging : time-domain or harmonic-domain ?

- **Time-domain** : imaging condition complicated but quite low computational cost
- **Harmonic-domain** : imaging condition simple but huge computational cost

Motivations

Imaging methods

- Full Wave Inversion (FWI) : **inversion process** requiring to solve many forward problems

Seismic imaging : time-domain or harmonic-domain ?

- **Time-domain** : imaging condition complicated but quite low computational cost
- **Harmonic-domain** : imaging condition simple but huge computational cost

Memory usage

Motivations

Resolution of the forward problem of the inversion process

- Elastic wave propagation in the frequency domain : Helmholtz equation

Motivations

Resolution of the forward problem of the inversion process

- Elastic wave propagation in the frequency domain : Helmholtz equation

First order formulation of Helmholtz wave equations

$\mathbf{x} = (x, y, z) \in \Omega \subset \mathbb{R}^3$,

$$\begin{cases} i\omega\rho(\mathbf{x})\mathbf{v}(\mathbf{x}) = \nabla\cdot \underline{\underline{\sigma}}(\mathbf{x}) + f_s(\mathbf{x}) \\ i\omega\underline{\underline{\sigma}}(\mathbf{x}) = \underline{\underline{C}}(\mathbf{x})\ \underline{\underline{\varepsilon}}(\mathbf{v}(\mathbf{x})) \end{cases}$$

- $\mathbf{v}$: velocity vector
- $\underline{\underline{\sigma}}$: stress tensor
- $\underline{\underline{\varepsilon}}$: strain tensor

Approximation methods

Discontinuous Galerkin Methods

- ✓ unstructured tetrahedral meshes
- ✓ combination between FEM and finite volume method (FVM)
- ✓ *hp*-adaptivity
- ✓ easily parallelizable method

Approximation methods

Discontinuous Galerkin Methods

✓ unstructured tetrahedral meshes

✓ combination between FEM and finite volume method (FVM)

✓ *hp*-adaptivity

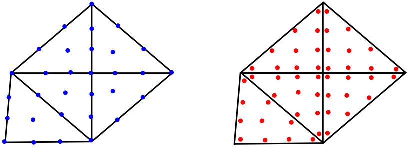
✓ easily parallelizable method

✗ ✗ large number of DOF as compared to classical FEM

Approximation methods

Discontinuous Galerkin Methods

✓ unstructured tetrahedral meshes
✓ combination between FEM and finite volume method (FVM)
✓ *hp*-adaptivity
✓ easily parallelizable method
✗ ✗ large number of DOF as compared to classical FEM



Approximation methods

Hybridizable Discontinuous Galerkin Methods

✓ same advantages as DG methods : unstructured tetrahedral meshes, *hp*-adaptivity, easily parallelizable method, discontinuous basis functions

✓ introduction of a new variable defined only on the interfaces

✓ lower number of coupled DOF than classical DG methods

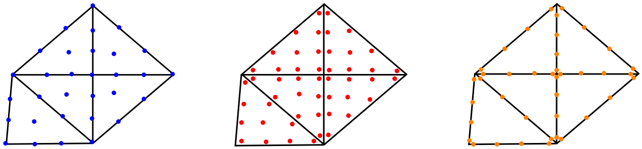
Approximation methods

Hybridizable Discontinuous Galerkin Methods

✓ same advantages as DG methods : unstructured tetrahedral meshes, *hp*-adaptivity, easily parallelizable method, discontinuous basis functions

✓ introduction of a new variable defined only on the interfaces

✓ lower number of coupled DOF than classical DG methods



Principles of the HDG method

1. Introduction of a Lagrange multiplier λ

Principles of the HDG method

1. Introduction of a Lagrange multiplier λ
2. Expressing the initial unknowns $\mathbf{W}^K = \left(\mathbf{v}^K,\ \underline{\sigma}^K\right)^T$ as a function of $\lambda = \left(\lambda^{F_1},\ \lambda^{F_2}, ...,\ \lambda^{F_{n_f}}\right)^T$

$$\mathbf{W}^K = \mathbb{K}^K \lambda$$

Principles of the HDG method

1. Introduction of a Lagrange multiplier λ
2. Expressing the initial unknowns $\mathbf{W}^K = \left(\mathbf{v}^K, \underline{\sigma}^K\right)^T$ as a function of $\lambda = \left(\lambda^{F_1}, \lambda^{F_2}, ..., \lambda^{F_{n_f}}\right)^T$

$$\mathbf{W}^K = \mathbb{K}^K \lambda$$

3. Transmission condition : $\displaystyle\sum_K \mathbb{B}^K \mathbf{W}^K + \mathbb{L}^K \lambda = 0$

Principles of the HDG method

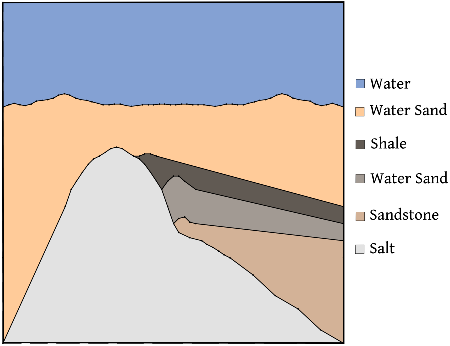
1. Introduction of a Lagrange multiplier λ
2. Expressing the initial unknowns $\mathbf{W}^K = \left(\mathbf{v}^K,\ \underline{\sigma}^K\right)^T$ as a function of $\lambda = \left(\lambda^{F_1},\ \lambda^{F_2}, ...,\ \lambda^{F_{n_f}}\right)^T$
$$\mathbf{W}^K = \mathbb{K}^K \lambda$$
3. Transmission condition : $\displaystyle\sum_K \mathbb{B}^K \mathbf{W}^K + \mathbb{L}^K \lambda = 0$
4. $\mathbb{M}\lambda = \mathbb{S}$

Principles of the HDG method

1. Introduction of a Lagrange multiplier λ
2. Expressing the initial unknowns $\mathbf{W}^K = \left(\mathbf{v}^K,\ \underline{\sigma}^K\right)^T$ as a function of $\lambda = \left(\lambda^{F_1},\ \lambda^{F_2}, ...,\ \lambda^{F_{n_f}}\right)^T$
$$\mathbf{W}^K = \mathbb{K}^K \lambda$$
3. Transmission condition : $\displaystyle\sum_K \mathbb{B}^K \mathbf{W}^K + \mathbb{L}^K \lambda = 0$
4. $\mathbb{M}\lambda = \mathbb{S}$
5. Computation of the solutions of the initial problem, element by element

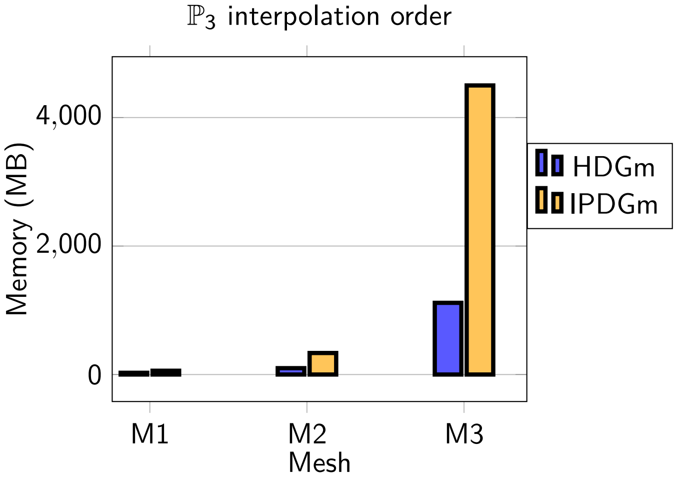
Contents

Anisotropic test case



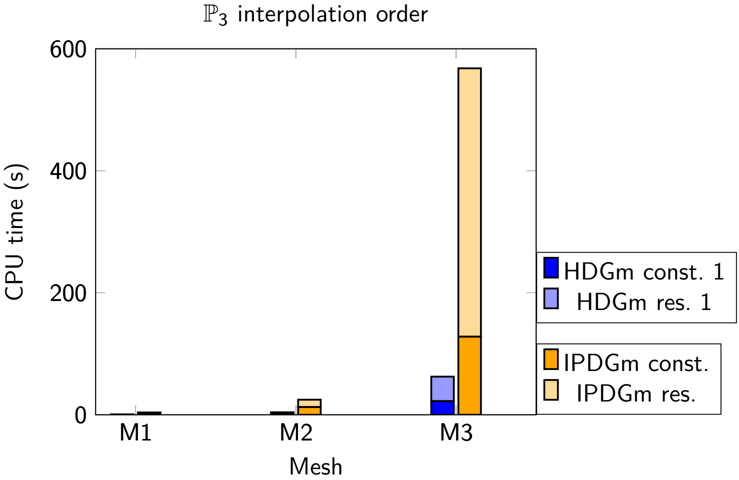
- Three meshes :
  - 600 elements
  - 3000 elements
  - 28000 elements

Anisotropic case : Memory consumption



$\textbf{IPDG}_{\textbf{memory}} \simeq \textcolor{red}{\textbf{4}} \times \textbf{HDG}_{\textbf{memory}}$

Anisotropic case : CPU time (s)



$$\mathbf{IPDG}_{\mathbf{CPUtime}} \simeq \mathbf{9} \times \mathbf{HDG}_{\mathbf{CPUtime}}$$

Contents

Resolution of the linear system $\mathbb{M}\lambda = \mathbb{S}$

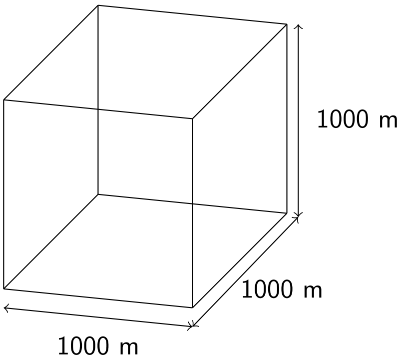
- **direct solver : MUMPS** (MUltifrontal Massively Parallel sparse direct Solver) :
  - Direct factorization $A = LU$ or $A = LDL^T$
  - Multiple RHS
- **hybrid solver : MaPhys** (Massively Parallel Hybrid Solver) :
  - combination of direct and iterative methods
  - non-overlapping algebraic domain decomposition method (Schur complement method)

Cluster configuration

Features of the nodes :

- 2 Dodeca-core Haswell Intel Xeon E5-2680
- Frequency : 2,5 GHz
- RAM : 128 Go
- Storage : 500 Go
- Infiniband QDR TrueScale : 40Gb/s
- Ethernet : 1Gb/s

3D plane wave in an homogeneous medium



Configuration of the computational domain Ω .

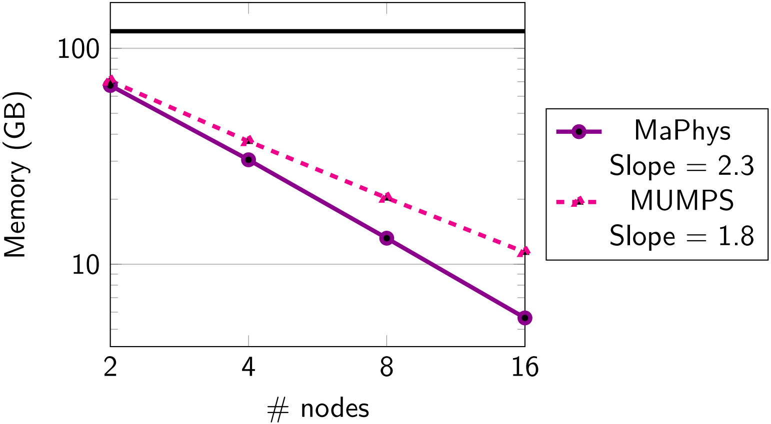
- Physical parameters :
  - $\rho = 1$ kg.m^{-3}
  - $\lambda = 16$ GPa
  - $\mu = 8$ GPa
- Plane wave :

$$u = \nabla e^{i(k_x x + k_y y + k_z z)}$$

- $\omega = 2\pi f, f = 8$ Hz
- Mesh : 21 000 elements

3D Plane wave : Memory consumption

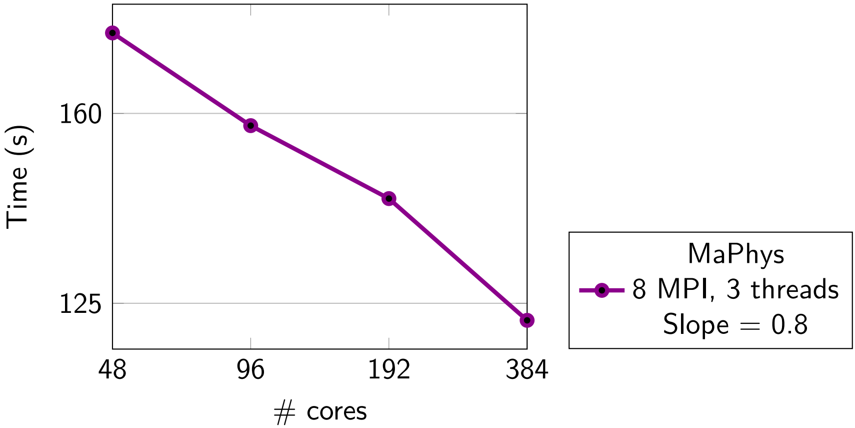
Average memory for one node (8 MPI by node and 3 threads by MPI)



(matrix order = 1 300 000, # nz=300 000 000)

3D Plane wave : Execution time

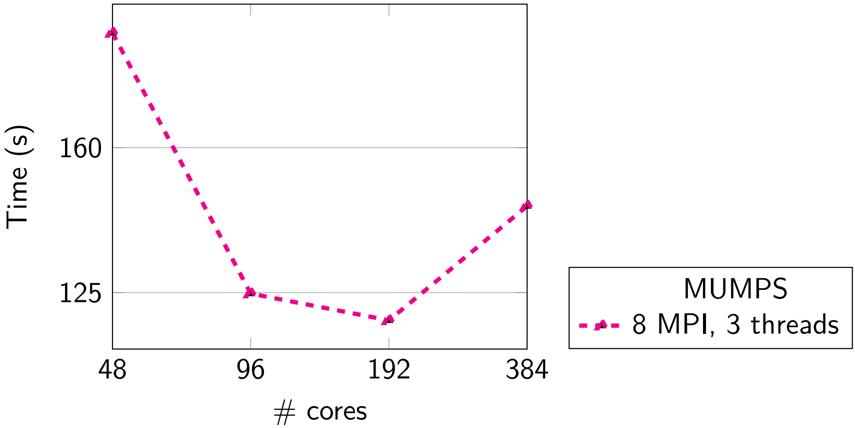
Execution time for the resolution of the HDG-$\mathbb{P}_3$ system



(matrix order = 1 300 000, # nz=300 000 000)

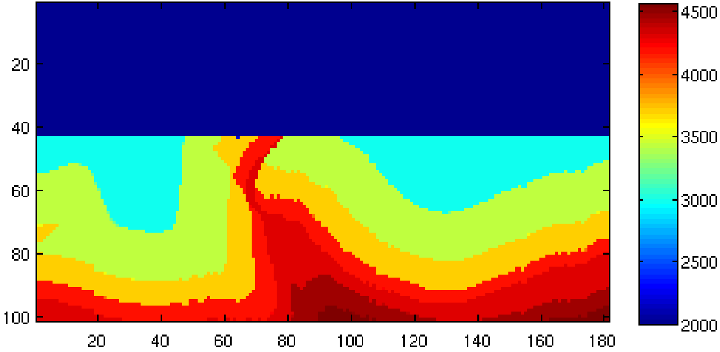
3D Plane wave : Execution time

Execution time for the resolution of the HDG-$\mathbb{P}_3$ system



(matrix order = 1 300 000, # nz=300 000 000)

Epati test-case

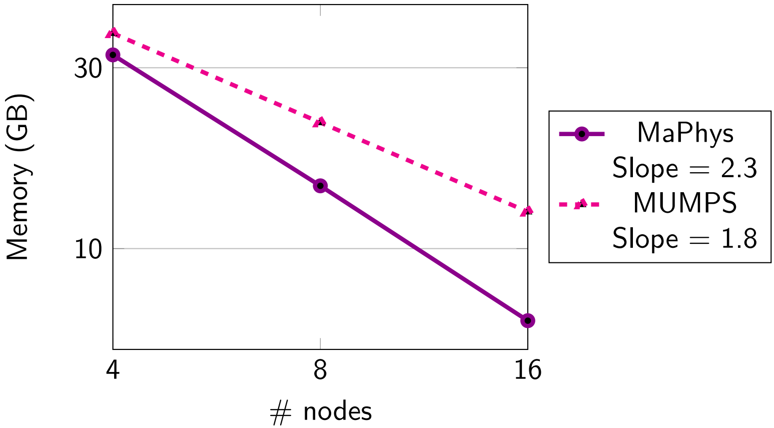


V_p-velocity model ($m.s^{-1}$), vertical section at $y = 700$ m

Mesh composed of 25 000 tetrahedrons

Epati test-case : Memory consumption

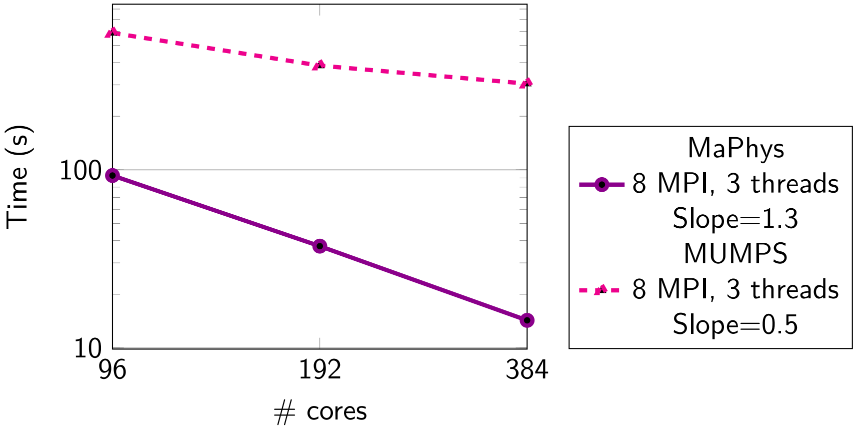
Average memory for one node (8 MPI by node and 3 threads by MPI)



(matrix order = 1 300 000, # nz=365 000 000)

Epati test-case : Execution time

Execution time for the resolution of the HDG-$\mathbb{P}_3$ system



(matrix order = 1 300 000, # nz=365 000 000)

Conclusion-Perspectives

- more detailed analysis of the comparison between solvers
  - larger meshes
  - more powerful clusters
- memory crash test
- extension to elasto-acoustic case

Thank you !